REMARKS

Status of the Claims

Claim 1 is currently presented in the Application and is an independent method claim. Claim 1 has been amended in this Response. Applicants are not conceding that the subject matter encompassed by claim 1, prior to Amendment, is not patentable over the art cited by the Examiner. Claim 1 was amended solely to facilitate expeditious prosecution of the Application. Applicants respectfully reserve the right to pursue claims, including the subject matter encompassed by claim 1, as presented prior to this Amendment, and additional claims, in one or more continuation and/or divisional patent applications.

Support for the amendments to claim 1 is found, for example, in Applicants' specification on page 47, line 25 through page 61, line 26 (also see Figures 43 through 53). No new matter has been added as a result of the amendments to claim 1.

Drawings

Applicants note that the Examiner has not indicated whether Applicants' formal drawings, filed with the Application on September 25, 2003, are accepted. Applicants respectfully request that the Examiner indicate whether Applicants' formal drawings are accepted in the next action.

Double Patenting Rejection

Claim 1 is provisionally rejected on the ground of non-statutory obviousness-type double patenting as being unpatentable over claim 1 of co-pending Application No. 10/670,835 in view of Washington et al., U.S. Patent No. 5,835,775 (hereinafter Washington). Applicants hereby submit a Terminal Disclaimer (see attached) and therefore respectfully request that the double-patenting rejection be removed.

Claim Rejections – Alleged Obviousness Under 35 U.S.C. § 103

Claim 1 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Washington. Applicants respectfully traverse the rejections under 35 U.S.C. § 103.

Washington purports to teach a method and computer system for executing family generic, processor specific files (see Washington, Abstract). Applicants respectfully submit that Washington does not teach or suggest several of the elements of Applicants' independent claim 1.

Applicants teach and claim managing a plurality of processors as virtual devices. A first processor receives a device request that "corresponds to a virtual device task." Data corresponding to the device request is stored in a common memory that is shared by a plurality of heterogeneous processors. The first processor creates a task block in the common memory, "the task block including a software code identifier that corresponds to a first device code module and an input buffer address." The first processor then signals the second processor, and the second processor reads the address of the task block and retrieves the software code identifier from the task block. The second processor uses the input buffer address to read data from the input buffer. Then the second processor determines "whether the first device code module corresponding to the software code identifier is loaded in the second processor's local memory, wherein the first device code module is capable of performing the first virtual device task." If the first device code module is not loaded in the second processor's local memory, the first device code module is read from the common memory into the second processor's local memory. Finally, the data is processed by the second processor "using the first device code module stored in the second processor's local memory, wherein the processing comprises executing the first device code module to perform the first virtual device task."

Applicants respectfully submit that Washington does not teach or suggest several elements of Applicants' independent claim 1. In contrast to Applicants' claimed invention, Washington discusses a first processor executing a processor test section which then "returns an identifier representing the processor type of the first processor" (Washington, col. 5, lines 50-52). Washington discloses producing one object/executable file that can be executed by an entire family of processors, rather than separate object files for different processors. Washington states that "[i]t is yet another object of the present invention to provide a method for ***generating a single program***

executable that is optimized ***for execution on at least two different processors***. It is yet a further object of the present invention to provide a method for ***generating a single program executable*** for a computer system having ***two processors with different instruction sets***" (Washington, col. 3, lines 16-22, emphasis added). According to Washington, once a processor begins executing code, the processor first executes a processor test section that identifies the processor type of the processor (Washington, col. 5, lines 45-59). Based on the processor type, certain sections of the executable file will or will not be executed (Washington, col. 5, line 60 through col. 6, line 11).

Applicants respectfully submit that Washington does not teach or suggest "loading a plurality of device code modules into a common memory" where "each device code module "performs a different virtual device task." Washington does not appear to be concerned with device code modules or virtual device tasks. Applicants further submit that Washington does not teach or suggest "receiving a device request at a task queue manager running on a first processor" where the device request that is received "corresponds to a first virtual device task." The cited sections of Washington (col. 3, lines 64-65; col. 5, lines 47-48; and col. 8, line 16) merely discuss receiving a request to execute the FGPS (family generic/processor specific application). The FGPS file includes a CPU test section, processor type flag, common code section, processor specific section, and family generic section. Receiving a request to execute the FGPS is not the same as receiving a ***device request*** that corresponds to a first ***virtual device task.***

Washington also does not teach or suggest "creating" a task block in the common memory, where the task block includes a "software code identifier" that "corresponds to a first device code module." The cited section of Washington (col. 7, lines 1-5) discusses a file type identifier included in a file header of an FGPS file. However, this file type identifier does not "correspond to a first device code module," where the first device code module is one of "a plurality of device code modules" loaded "into a common memory of a computer system, wherein each device code module performs a different virtual device task when executed," as taught and claimed by Applicants. Applicants teach and claim that each device code module corresponds to a

different virtual device task. The "file type identifier" discussed in col. 7, lines 1-5 of Washington is merely said to be "an identifier representing a FGPS file type," and does not appear to have anything to do with device code modules or virtual device tasks. Nor does Washington teach or suggest the first processor "signaling" the second processor by "writing the address of the task block to a mailbox corresponding to the second processor." Washington does not mention any type of "mailbox" system. The cited section of Washington merely discusses writing an address of an instruction block to a memory area. Washington does not appear to perform any type of "signaling" to a "mailbox" as taught and claimed by Applicants.

Furthermore, Washington does not teach or suggest "determining whether the first device code module corresponding to the software code identifier is loaded in the second processor's local memory, ***wherein the first device code module is capable of performing the first virtual device task,***" and then "processing the data by the second processor using the first device code module," where "***the processing comprises executing the first device code module to perform the first device task***." Applicants teach and claim that a processor can act as a virtual device by executing a device code module that will perform a requested device task. The cited sections of Washington at col. 5, lines 44-48 merely discuss receiving a request to execute the FGPS file and then executing the FGPS file. This is not analogous to loading and executing ***a device code module that is capable of performing a virtual device task***. Washington does not disclose using a processor to act as a virtual device, as taught and claimed by Applicants.

Based on the above, Applicants respectfully submit that independent claim 1 is patentable over Washington, and respectfully request that it be allowed.

Conclusion

As a result of the foregoing, it is asserted by Applicants that the remaining claims in the Application are in condition for allowance, and Applicants respectfully request an early allowance of such claims.

Applicants respectfully request that the Examiner contact the Applicants' attorney listed below if the Examiner believes that such a discussion would be helpful in resolving any remaining questions or issues related to this Application.

Respectfully submitted,

By /Leslie A. Van Leeuwen, Reg. No. 42,196/

Leslie A. Van Leeuwen, Reg. No. 42,196
Van Leeuwen & Van Leeuwen
Attorneys for Applicant
Telephone: (512) 301-6738
Facsimile: (512) 301-6742